GaMuSo: Graph base Music recommendation in a Social bookmarking service.

Jeroen De Knijf[1], Anthony Liekens[2], and Bart Goethals[1]

[1] Department of Mathematics and Computer Science, Antwerp University
[2] VIB Department of Molecular Genetics, Antwerp University

Abstract. In this work we describe a recommendation system based upon user-generated description (tags) of content. In particular, we describe an experimental system (GaMuSo) that consists of more than 140.000 user-defined tags for over 400.000 artists. From this data we constructed a bipartite graph, linking artists via tags to other artists. On the resulting graph we compute related artists for an initial artist of interest. In this work we describe and analyse our system and show that a straightforward recommendation approach leads to related concepts that are overly general, that is, concepts that are related to almost every other concept in the graph. Additionally, we describe a method to provide functional hypothesis for recommendations, given the user insight why concepts are related. GaMuSo is implemented as a webservice and available at: music.biograph.be.

1 Introduction

The last decade, social bookmarking services have emerged as a valuable tool to collectively organize online content. Well known examples of such services are the popular photo-sharing site Flickr, del.icio.us a webservice to share bookmarks of the WWW, CiteUlike that let users store and tag scholarly articles, and last.fm a webservice focused on describing bands and artists. The common feature that these diverse webservices share is that they provide users with the ability to tag resources. Tags are freely chosen keywords and provide a simple tool to organize, search and explore the content. From an intelligent data analysis point of view, these so called social bookmarking resources are extremely precious to analyse and predict complex real live resources. For example, in case one wants to construct a video or audio recommender system, the major problem is obtaining a real life dataset on which the recommendations can be based. Specifically, these data is often property of commercial companies and is due to privacy regulations and strategic business interest not freely available. For example, the well-known webservice Youtube stores user profiles containing videos that the user watched. Based upon this watching behavior Youtube recommend videos of interest to the users [2]. Likewise, recommendation systems of Amazon and Itunes are based on purchasing behaviour of the customers on the respective webstore. This in contrast with social bookmarking sites, where the user-defined tags are—in general—property of the respective users. Moreover, the goal of

these sites is to provide users with tools to share and discover content of interest, thus ensuring that user-defined tags are freely accessible to their respective community.

Despite the many benefits of collective tag based systems, collaborative tagging services also introduce serious challenges in order to analyze these resources. That is, because users are freely to chose tags, irrelevant, subjective, fraudulent and erroneous tags often occurs in collective tag based systems [1]. For example, for the last.fm tagset, we found—among many others—the following tags for U2: I Like, britpop, american band, The Moonlight Experience. While the first tag is obviously irrelevant and the britpop tag might give rise to discussion, the 'american band' tag is unmistakable wrong. The last tag is most likely a fraudulent type of tag to boost the popularity of an unknown band (11 people listened TME on last.fm).

In this work we describe our system (GaMuSo) for discovering 'similar' concepts in a social collaborative tagging service and analyze it on the well known collaborative tagging service for artists and bands: last.fm. In particular, our analysis is concerned with the following question: given an artist/band, which artists/bands are most similar to the given one ? Besides the recommendation, we also derive hypothesis why the two concepts are related. As such, our system does not work as a black box, but provides the user with background information on the recommendation. In order to perform the recommendation, we transform the derived tags/artists into a directed bipartite weighted graph. On this graph artists are only linked with other artists via user-defined tags. The recommendation algorithm of GaMuSo is based upon random walks and random walks with restart. We argue theoretically and demonstrate in our experiments that the standard method in the data mining literature to derive related nodes in a graph (see for example [9]) is likely to derive overly general related nodes. We propose an algorithm to compensate for this flaw, effectively enforcing GaMuSo to derive more specific similar concepts.

In the next section we describe the tagset used in more detail, and report on the data cleaning and graph construction from this dataset. In section 3 we describe random walks, random walks with restart and our method to derive similar concepts. Moreover, in this section we describe the heuristic to derive functional hypothesis for the recommendation. In the following section, we extensively evaluate the different approaches and discuss the results. In section 5 we describe related work. Finally, in section 6 we draw conclusions and give directions for further research.

2 Graph Construction

From Last.fm we retrieved 443.816 names of artists and 127.516 tags describing these artists. For every artist, we have all the user-defined tags that are used to label this artist. Moreover, the tags for an artist are normalized, and given a weight relative to the most popular tag for the artist. These weights (between 1 and 100) correspond to the number of (distinct) users that assigned the tag to

the artist. In particular, the most popular tag for an artist is assigned the weight 100, and all other tags are weighted in accordance with their frequency relative to the most frequent tag. The same weighting holds for the tag-to-artist relation. Note, however, that the weight for the artist-to-tag relation is in general different from the weight for the same reversed relation. This is the case, because in the former case weights are normalized per tag, while in the latter case weights are normalized per artist.

The next step consists of data cleaning: tags or artists with the same label, but different in capital letters/lowercase letters, punctuation, spacing etc. are transformed into uniform writing style. To do so, we removed all non-alpha and non-digits characters from the labels, transformed all capital characters into lowercase ones, replaced "&" by "and" and removed the definite article "the" from the beginning of the label. As a result, Beatles and The Beatles are the same, as well as post-modernism, postmodernism and post modernism. The resulting dataset consists of 109.345 tags and 407.036 artists.

Finally, we map the cleaned data into a weighted directed graph. A graph $G = \{V, E, \lambda\}$ is defined as a directed weighted graph, where V is a set of vertices, $E \subseteq V \times V$ a set of edges (ordered pair of vertices) and λ a labeling function that assigns a weight from the interval $[1, 100]$ to each edge, i.e. $\lambda : E \rightarrow [1, 100]$. Besides the definition of directed weighted graph, we also needs the notion of a simple path in a directed graph. A simple path is a path containing no repeated vertices. More formally: a simple path in G is a sequence of nodes $P = v_1 v_2 \dots v_n$ such that for all $1 \leq k < n : (v_k, v_{k+1}) \in E$. Moreover, for all $1 \leq j < k \leq n$ it holds that $v_j \neq v_k$.

Every artist and every tag is mapped to a distinct node in the graph. The weighted directed edges in the graph correspond to the relations from tag-to-artist and from artist-to-tag. In our setting the weight of the edge is equal to the respective weight of the relation. Finally, as a last pre-processing step we removed all nodes that had less than two outlinks in the graph, effectively removing concepts that where badly connected. The resulting graph consists of 49.022 tags, describing 18.634 artists. In comparison with the original dataset, it is especially for the number of artists a huge reduction. The main reason for this is that a lot of artist are simply not tagged.

3 Algorithm

Finding related nodes in a graph, is in the data mining literature often solved by using random walk with restart [9,8]. In this section we argue that such an approach leads to related concepts that are overly general, and propose an adjustment to overcome this deficit. First, we describe random walks and random walks with restart on directed graphs. Then we propose our algorithm. Finally, we describe an heuristic to generate functional hypotheses for concepts that are assumed to be related.

3.1 Random Walks & Random Walk with Restarts

Informally, a random walk on a directed weighted graph G can be described as follows: consider a random particle currently at node v in G, at each iteration the particle moves to a neighbor—via an outgoing edge—with a probability that is proportional to its edge weight. The steady state probability of a node w, denoted as $\mathbf{u}(w)$ states the probability that the particle visits this node in the long term. Intuitively, the steady state probability of node w states the importance of w in the graph, i.e. $\mathbf{u}(w)$ is also known as the eigenvector centrality of w. In order for a random walk to converge to the steady state probability, its underlying transition graph must be ergodic. A standard method [6] of enforcing this property for random walks on directed graphs, is to add a new set of complete outgoing transitions, with small transition probabilities, to all nodes, creating a complete (and thus an aperiodic and strongly connected) transition graph. Hence, instead of moving to a neighbor with a probability that is proportional to its edge weight, the particle also has small probability of moving uniformly at random to any node in the graph.

The steady state probability of a random walk, can easily be computed by iteratively applying Equation 1, this method is also known as the power iteration method. In this equation M_G equals the column normalized adjacency matrix of G, $\boldsymbol{g}$ the normalized restart vector where for each node of G its corresponding value in $\boldsymbol{g}$ is set to 1, and c the restart probability for the uniform restart vector (also known as the damping factor).

$$\boldsymbol{u_{k+1}} = (1 - c) \times M_G \times \boldsymbol{u_k} + c \times \boldsymbol{g} \tag{1}$$

Given an initial set of nodes of interest $R \subseteq V$, in addition to a random walk, a random walk with restart has an additional probability d of jumping back to one of the nodes in R. The relevance score of node w with the set of nodes R ($\boldsymbol{u}_R(w)$) is then defined as the steady state probability of the random walk with restart to R. Note that, as in the previous case, a set of transition probabilities to all nodes in the graph is added. Hence, two sets of transition probabilities are added to the transition graph: one—to ensure convergence—with transition probability $c/|V|$ to all the nodes of the graph and one with transition probability $d/|R|$ to all the nodes of the initial set of interest R.

As in the previous case, the steady state probability of a random walk with restart can be computed by iteratively applying, a slightly adjusted version of, the power iteration method, shown in Equation 2.

$$\boldsymbol{u}_R^{k+1} = (1 - (c + d)) \times M_G \times \boldsymbol{u}_R^k + (c \times \boldsymbol{g}) + (d \times \boldsymbol{r}) \tag{2}$$

With M_G the column normalized adjacency matrix of G, c the restart probability to all nodes in the graph, $\boldsymbol{g}$ the restart vector to all nodes in the graph, d the restart probability to the interesting nodes in the graph (i.e. R) and $\boldsymbol{r}$ the restart vector for the nodes in R.

3.2 Finding Related Nodes

Most graph mining algorithms based upon random walks with restart (see for example [9,8]) solely use the steady state probability of the random walk with restart as the relevance score between nodes. However, consider the following example: Given two nodes w and v, and a set of restart nodes R. Suppose that $\boldsymbol{u}_R(w) = 0.2$ and $\boldsymbol{u}_R(v) = 0.3$. With this outcome, the most related node to the nodes of R is v. Now suppose that, $\boldsymbol{u}(w) = 0.1$ and $\boldsymbol{u}(v) = 0.6$. Hence, the apriori relevance of node v is much higher than the relevance score of node v with respect to the set of nodes R. In fact the initial set R harms the importance of v, while node w becomes far more important due to the initial set R. But nevertheless, when only using the steady state probability of the random walk with restart, v is preferred above w. In GaMuSo we take the prior importance of a node into account to adjust the score of the random walk with restart. In particular, for an initial set R, the relevance score of a node v is determined by

$$\frac{\boldsymbol{u}_R(v)}{\boldsymbol{u}(v)}. \tag{3}$$

Intuitively, this adjustment lowers the probability of objects that are similar to most other objects, effectively enforcing the recommendation to be more specific. In the experimental setting we discuss the results obtained with and without the adjustment.

Note that in our setting, the set of restart nodes always consists of one node. Moreover, we fixed the restart probability c to all nodes in the graph to 0.05 and the restart probability d for the set of nodes of interest , i.e. R, to 0.25. In general, different values for these probabilities results in minor changes in the recommendation, as long as $d >> c$.

3.3 Automatic Hypotheses Generation

The last part of GaMuSo consists of a method for automatic hypothesis generation. That is, between a source and a target node (e.g. artist of interest and related artist) a set of most likely, simple paths are generated. Most likely, in the sense that from all simple paths between source and target, the aim of the greedy heuristic is to select those that have the highest probability that the random walker traversed over this path between source and target.

Assume a source node s and a target node t in G. Let $P = v_1, \ldots, v_n$ be a simple path between source and target, with $v_1 = s$ and $v_n = t$. Then, the probability of a random walker to traverse this path ($\Pi(P_{s,t})$) provided it starts in s and end in t equals:

$$\prod_{i=1}^{n-1} \mathrm{p}(v_i, v_{i+1}). \tag{4}$$

with $p(v_i, v_{i+1})$ the probability of moving from v_i to v_{i+1}, that is the transition probability from v_i to v_{i+1}.

Likewise, for a path P' starting in some intermediate node $v_i \neq s$ and ending in t, we estimate the probability that the random walk form source to target traversed this path with:

$$\boldsymbol{u_s}(v_i) \times \Pi(P'_{s,t}). \tag{5}$$

where $\boldsymbol{u_s}(V_i)$ is the steady state probability of the random walk with restart to the source node (Equation 2).

In order to find the k most probable paths from source to target, we start from the target node and use the estimate of Equation 5 to find the most likely paths of length two. Consequently, we recursively extends these paths until we found the k most likely paths connecting source to target. At each iteration we only keep the K, with $K >> k$, most likely partial paths (in accordance with the estimate derived in Equation 5), effectively reducing the search space to a workable amount. The pseudo code is given in Algorithm 1. In our experiments we set the number of paths to ten (i.e. $k = 10$) and pruned the search space when there where more than 1000 paths under consideration ($K = 1000$).

Algorithm 1 Path Generation

Input: graph $G = \{V, E, \lambda\}$, source node s, target node t, number of paths k
Output: list of k most likely paths between s and t

$S \leftarrow \{\{t\}\}$
repeat
  $S' \leftarrow \{\}$
  for all paths $P = \{v_1 \dots v_n, t\} \in S$ **do**
    if $v_1 == s$ **then**
      $S' \leftarrow S' \bigcup P$
    else
      for all $n : (n, v_i) \in E$ and $n \notin \{v_1 \dots v_n, t\}$ **do**
        $S' \leftarrow S' \bigcup \{n, v_1 \dots v_n, t\}$
      end for
    end if
  end for
  $S \leftarrow$ prune S' to K most likely paths
until there at least k paths in S from s to t
return top k paths in S

4 Experiments

In this section we describe the experiments of our artist recommendation system on the last.fm dataset. Although, determining the most related artists for a given artist is a rather subjective opinion, we will evaluate the experiments by commenting on the results and by investigating the generated hypotheses for

1	Radiohead	6	Arcade Fire
2	The Beatles	7	David Bowie
3	Muse	8	Coldplay
4	The Killers	9	The Red Hot Chili Peppers
5	Björk	10	Pink Floyd

Table 1. The 10 most important bands in the last.fm network.

the experiments. Moreover, we compare GaMuSo with a purely random walk with restarts based approach.

The first experiment consists of finding the most important artists in the graph, that is the artist with the largest eigenvector centrality. This measures can be computed using Equation 1. The ten most important artists in the last.fm network are displayed in Table 1. Remarkable is that most artists can best be described by the genre "alternative rock", and that today's most popular artists (according to the billboard charts, such as Justin Bieber, Lady Gaga, etc) are missing. Moreover, complete different music styles like classical music or Jazz are also absent in the top ten.

Our next experiment consist of deriving the most related artists for The Stooges, a well known American punk band from the late sixties and early seventies. In this experiment we compare the results of the recommendation by GaMuSo with the results from a purely random walk with restart based recommendation, i.e. using only Equation2. The results are shown in Table 2. The first remarkable observation is that the results for the RWR approach consist mainly of well known bands, while the results of our recommendation service consists—except for the connoisseur of this genre—of unknown bands. A related observation is that five out of ten recommendations from the RWR approach belong to the top ten most important nodes in the last.fm network. This observation supports our claim that a pure RWR based approach for finding related concepts can result in overly general related results.

Another sanity check to judge the predictions can be achieved by examining the most likely paths between The Stooges and the most related prediction by GaMuSo: MC5. These ten most likely paths are shown in Figure 1. From

The Beatles	MC5
Radiohead	Modern Lovers
Green Day	Iggy Pop & James Williamson
David Bowie	Richard Hell and the Voidoids
The Rolling Stones	Mink DeVille
Muse	Flamin' Groovies
The Strokes	The Monks
Kings of Leon	? and the Mysterians
The White Stripes	New York Dolls
The Killers	The Sonics

Table 2. The 10 most related artists for The Stooges, according to a pure RWR based approach (left) and according to GaMuSo.

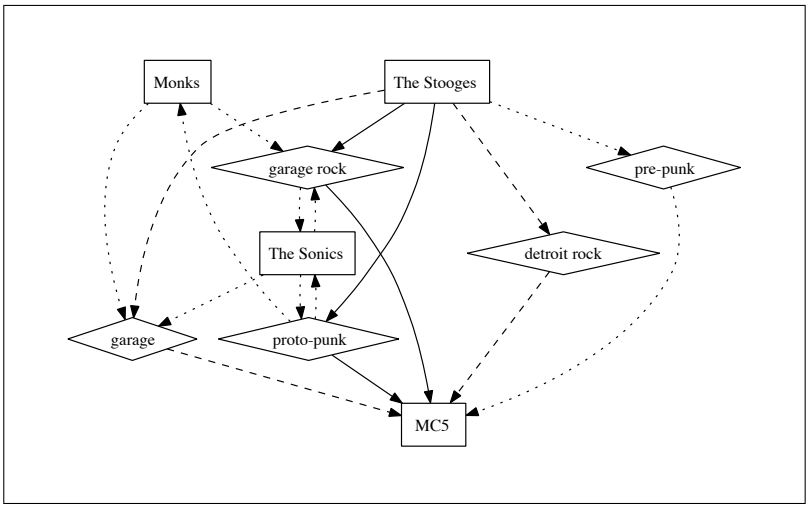

Fig. 1. The Stooges connect to MC5. Rectangles represent artist while diamonds represents user-defined tags.Full, dashed and dotted lines represent relations with decreasing probabilities.

these paths one can derive that the most influential tags for the prediction are proto-punk and garage rock, which make perfectly sense for The Stooges. Other influential tags are garage and detroit rock, while pre-punk is the least influential tag. Also some other top predicted bands are influential for the relation between the two concepts: The Monks and The Sonics, which both have common connections to the tags garage, proto-punk and garage rock.

Our following experiment is in the classical music genre; we retrieved the ten most related artists for the classical composer Frédérique Chopin. The results are shown in Table 3, where the left column contains the results for the RWR approach and the right column contains the results of GaMuSo.

The first noteworthy observation, is that the most similar results obtained by GaMuSo are all but one known as composers of the Romantic music era.

Ludwig Van Beethoven,	Felix Mendelssohn
Ludovico Einaudi	Gabriel Fauré
Philip Glass	Robert Schumann
Radiohead	Sir Edward Elgar
Erik Satie	Franz Schubert
Pyotr Llyich Tchaikovsky	Richard Wagner
Claude Debussy	Ron Pope
Yann Tiersen	Edvard Grieg
Howard Shore	Giuseppe Verdi
Antonín Dvořák	Johannes Brahms

Table 3. The 10 most related artists for Frederique Chopin, according to a pure RWR based approach (left) and according to GaMuSo (right).

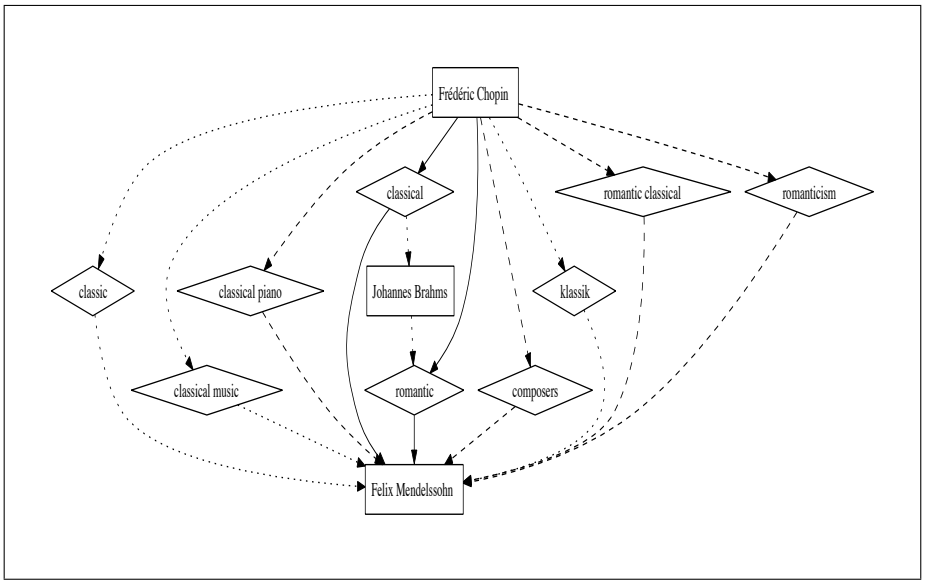

Fig. 2. Frédérique Chopin connect to Felix Mendlssohn. Rectangles represent artist while diamonds represents user-defined tags.Full, dashed and dotted lines represent relations with decreasing probabilities.

Note that, Chopin is considered as one of the most influential composers of Romantic music. The only composer that is not from the Romantic era is Ron Pope, who is a modern composer of classical piano music. Further noteworthy is that two of the related composers (Schumann and Brahms) are—according to Wikipedia [11]—greatly influenced by Chopin.

Also when examining the functional hypotheses between Chopin and Mendlssohn, it is clear that the tag romantic plays an important role in the random walk from Chopin to Mendlssohn. Besides the romantic tag, also the classic tag and its many variations are of great influence. Further interesting is the relatively high impact of the tags composer and classical piano.

Further investigations into the predictions posed by the RWR approach (Table 3) reveals that only three out of nine suggested composer are from the Romantic era. Three out of nine are composers from the $20Th$ century (Philip Glass, Yann Tiersen, Howard Shore), and have in comparison with earlier classical composers as Beethoven and Mozart, a prior importance in the last.fm graph that is almost ten times higher. Apparently, it is the case that these $20Th$ century composers are far more popular for the last.fm audience. Another remarkable recommendation is the high similarity between Radiohead and Chopin. Radiohead is the most important artist in the last.fm network(see Table 1), and is a well known alternative rock/indie band. Examining the paths between Chopin and Radiohead reveals that practically all paths go over the tag piano, which is an adequate tag for Chopin but seems less relevant for Radiohead. Finally,

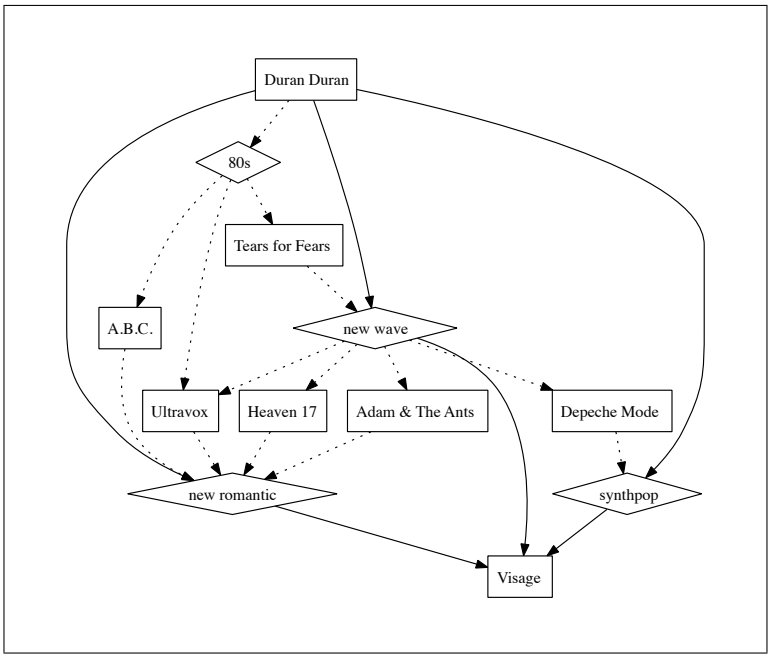

Fig. 3. Duran Duran connected to Visage. Rectangles represent artist while diamonds represents user-defined tags.Full, dashed and dotted lines represent relations with decreasing probabilities.

there are some good arguments to show Beethoven as one of the the most related composer to Chopin. That is, Beethoven is considered as the crucial figure in transition of the classical music into the Romantic era. However, Beethoven does not occur in the top ten recommendations of GaMuSo.

As a last experiment we derived the most likely paths of a well known eighties band to one of its top predictions. The connection between Duran Duran and Visage is shown in Figure 3. Also in this case, it seems a sensible prediction. The most influential tags: new wave, new romantic and synth pop forms a good description of both Duran Duran as well as Visage. Moreover, the paths over the different $80's$ bands feeds the hung that this is a good prediction, because most band are somewhat related to both Visage as well as Duran Duran.

5 Related Work

Most existing work on mining tag based social bookmarking services, formulate their research question and apply their algorithm on the social bookmarking service for images: Flickr. Although, there are many similarities between the different social bookmarking services, a research question as finding the most similar images to an image, of for example the Eiffel tower, seems to be pointless. That is, for an image either it contains the Eiffel tower or it does not. As

such, different degrees of similarity, as in the case of social bookmarking sites for artists, scholarly papers and webpages, seems to apply to a far lesser extend for images.

A popular research question is the one of tag recommendation: given a set of tags for a certain concept, which tags are further appropriate for it ? Sigurbjörnsson et al. [7] propose a co-occurrence based method for tag recommendation. Krestel et al. [5] propose a tag recommendation system base upon a Latent Dirichlet Allocation model. Another commonly posed research question, is of finding groups—often as complete as possible—of similar concepts or tags. The work by Van Leeuwen et al. [10], tries to accomplish this task by mining frequent tagsets and select the most promising ones according to the MDL principle.

Related work to our hypotheses generation algorithm is the work by Faloutsos et al. [3] and the work by Hintsanen et al. [4]. The purpose of both methods is to find the best connecting subgraph between a source and target node. The main difference is, that the aim of our method is to select the paths that have the highest influence on the random walk from source to target.

6 Discussion and Conclusion

In this paper we described the GaMuSo system, a recommendation system based upon tagged content, created in social bookmarking services. Despite the many disadvantages of social bookmarking services, it has great potential usefulness for the intelligent data analysis community. This is mainly due to the wide public availability of user created tags, and the different types of—often complex—content that is being tagged. One of the foremost disadvantages concerned with analysing tagsets of social bookmarking services, are the inherent linguistic problems that comes with user-generated tagsets. Our experiments revealed that—in spite of the data cleaning—many syntactical variations for the same tag exists. For example, the tags: classic, classical, klassik and classical music all occurred in one experiment. It is our aspiration to develop advanced data cleaning techniques to tackle this problem in the GaMuSo system.

One of the main features of GaMuSo consists of a recommendation engine that forces the related concepts to be more specific than the recommendation based upon the commonly used RWR approach. In the experiments, we showed that this adjustment often leads to considerably better results. However, as noticed in the experiments of finding related composers to Chopin, one can think of particular settings in which more general concepts are preferred over specific ones. For example, in a recommendation system for scholarly papers, it make sense to take the experience of the user into account. That is, for beginning researchers, such as a Master or a starting PhD student, more general recommendations would be more appropriate. On the other hand, very specific recommendations for experienced researcher are also desirable. But also when considering music recommendation, these different settings of a recommenda-

tion system make sense. For example, for people who are exploring a new music genre, general recommendations seem more useful than specific ones. This is an open issue in the GaMuSo system, for which further research is needed in order to take these different scenarios into account. Another salient property of GaMuSo is its ability to provide functional hypotheses why two concepts are related. These hypotheses have proven to be useful in order to investigate why Chopin and Radiohead were related. But also in other recommendations where these hypotheses valuable.

In the experiments we focused on artist recommendation, however, in principle the GaMuSo system is also suited as a recommendation engine for different types of tagged content. However, an important requirement is that different degrees of similarity between the content exists. As previously argued, this is a suitable setting for social bookmarking sites of artists, scholarly papers and webpages and seems to apply to a far lesser extend when the content consists of images.

References

1. M. Ames and M. Naaman. Why we tag: motivations for annotation in mobile and online media. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, pages 971–980, 2007.
2. J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath. The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 293–296, 2010.
3. C. Faloutsos, K. McCurley, and A. Tomkins. Fast discovery of connection subgraphs. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, 2004.
4. P. Hintsanen and H. Toivonen. Finding reliable subgraphs from large probabilistic graphs. *Data Min. Knowl. Discov.*, 17:3–23, August 2008.
5. R. Krestel, P. Fankhauser, and W. Nejdl. Latent dirichlet allocation for tag recommendation. In *ACM conference on Recommender systems*, RecSys '09, 2009.
6. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.
7. B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *ACM Conference on World Wide Web*, WWW '08, 2008.
8. J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *IEE International Conference on Data Mining*, pages 418–425, 2005.
9. H. Tong, C. Faloutsos, and J. Pan. Random walk with restart: fast solutions and applications. *Knowl. Inf. Syst.*, 14(3):327–346, 2008.
10. M. van Leeuwen, F. Bonchi, B. Sigurbjörnsson, and A. Siebes. Compressing tags to find interesting media groups. In *ACM Conference on Information and Knowledge Management*, CIKM '09, 2009.
11. Wikipedia. Frédéric Chopin — Wikipedia, the free encyclopedia, 2011. [Online; accessed 23-July-2011].